

# Detalle de Desarrollo del Video Juego “Travel Galaxy”

A continuación, hare un breve repaso por el proceso de desarrollar el video juego solicitado en el tp II para la materia de Programación II. El mismo solicita que se realice un video juego desarrollado en lenguaje Python usando tanto la librería Pygame como así también el paradigma de programación orientada a objetos. También se solicita en el mismo que se modularice el programa a fin de establecer un mejor orden de código y funcionalidad. En este caso la temática es a libre elección y se solicita un nivel de juego.

## Elección de la temática

Como temática he decidido desarrollar un video de shooting, el cual estará contextualizado en una aventura espacial. El juego consistirá en una nave espacial que tiene que enfrentarse, en el espacio, a un conjunto de meteoritos a los cuales tendrá que eliminar por medio de disparos laser tanto para sumar puntuación, como para defenderse de los mismo y poder encontrar gracias a la destrucción de los mismos una de las ocho gemas espaciales. El nombre del video por lo tanto paso a ser “Travel Galaxy”.

## Diseño del nivel

### Nave

La idea es desarrollar un doble comando para el uso de la nave espacial, el mismo estará conformado tanto por el manejo de las teclas arriba, abajo, derecha e izquierda del teclado y podrá rotar la puntería de la nave con la ubicación del mouse, como así también disparar clickeando con el botón izquierdo del mismo.

### Asteroides

Como la intención del juego es que haya un uso multidireccional del jugador por toda la pantalla, lo que se propone en este caso es que los asteroides surjan no de arriba ni de un costado de la pantalla, sino que emerjan del medio de la pantalla teniendo un comportamiento random de sus direccionamientos. Con esto lo que se busca es que el usuario tenga que necesariamente usar tanto la mano izquierda para esquivar los objetos que puedan colisionar con la nave, como así también la mano derecha para apuntarles a los mismos objetos.

### Gema

La gema es un elemento de instancia final en el nivel ya que solo va a hacer su aparición cuando el usuario allá logrado alcanzar un determinado puntaje destruyendo asteroides. A su vez esta misma tendrá que ser recolectada por la nave para finalizar con el nivel.

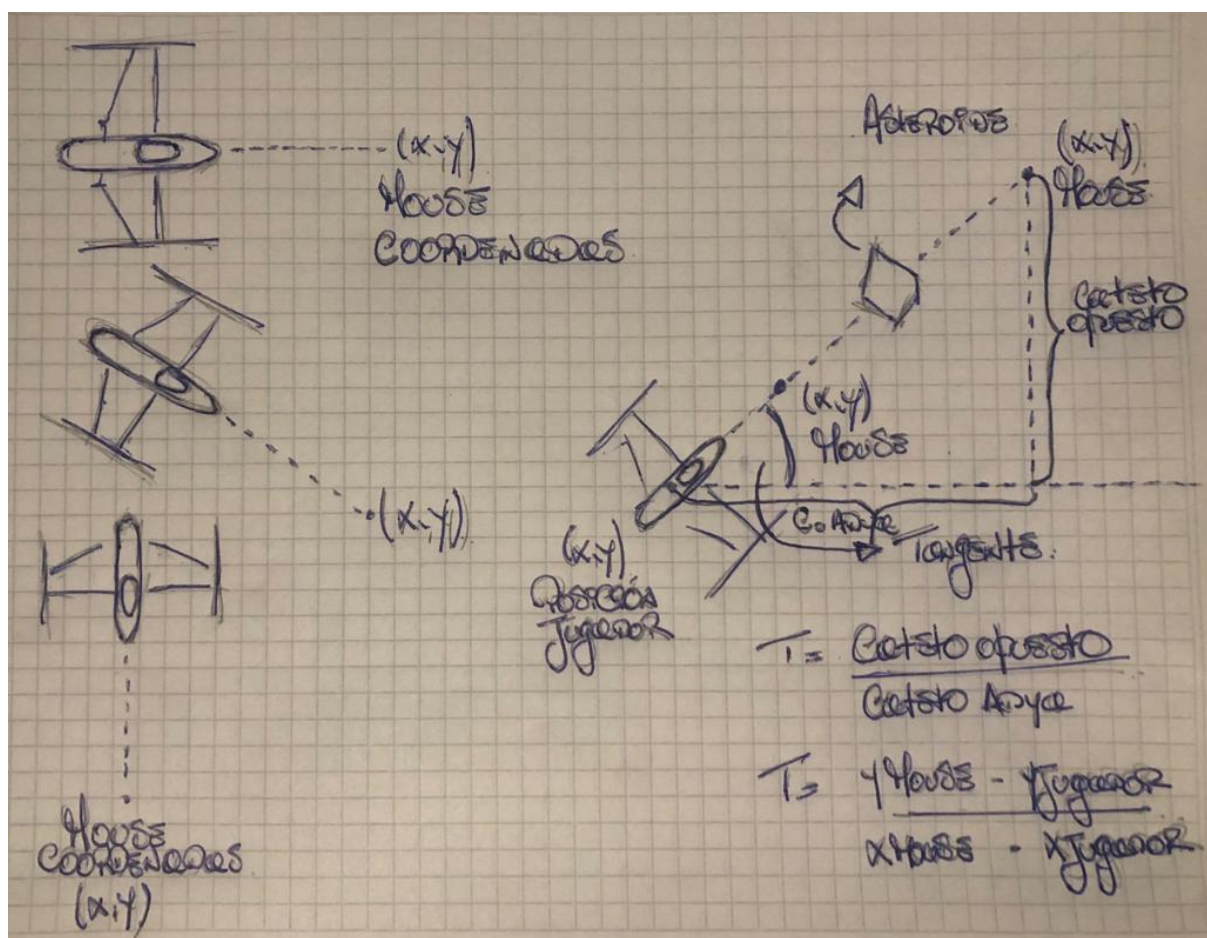
## Decisiones de diseño del juego

Al momento de pensar el juego se le dio mucha atención al uso de la espacialidad, por lo que tanto la creación de un doble comando para el manejo y uso de la nave, como el desarrollo de movimientos random de los asteroides y la aparición de la gema al final para su recolección tienen un claro objetivo.

Quiero resaltar la implementación de la Gema como elemento de importancia, no solo por que marca la finalidad del juego, sino por cómo define la dificultad del nivel. La particularidad de este elemento es que va a parecer en la posición en la cual se haya destruido el ultimo asteroide necesario para llegar a una determinada puntuación. Esto si bien no parece representar un problema, podría serlo teniendo en cuenta de cómo se esté logrando la destrucción de los asteroides ya que si el usuario se dedica simplemente a apuntar al centro de donde salen todos ellos, cuando llegue el momento de ir a recolectar la gema le será desde muy difícil hasta casi imposible. Con esta decisión lo que se busca es premiar que el usuario haya desarrollado un uso más amplio de la puntería y lo obligue, de alguna manera, a desplazarse en el espacio tanto esquivando asteroides como así también destruyéndolos tanto para su supervivencia como así también para la suma de puntos y superación del nivel.

## Detalles Técnicos Relevantes sobre la construcción del video Juego

Unos de los pilares del video es la posibilidad de maniobrar al personaje tanto en direcciones clásicas como así también en rotación sobre su propio eje, lo cual genera una amplia gama de movimientos además de exigirle al usuario una mayor concentración. A continuación, se muestran algunos bocetos (pasados a limpios) al momento de analizar este punto del video juego, además de que se explica parte relevante del código que permite esta implementación.



En esta imagen se analizar la rotación que se quiere tener disponible sobre el personaje y como con una función trigonométrica sencilla podemos obtener el ángulo de rotación para el mismo. Veamos un poco como se traslada esto al código.

```
def rotacion(self):
    self.mouse_coordenadas = pygame.mouse.get_pos()
    self.variacion_mouse_x = (self.mouse_coordenadas[0] - self.hitbox.centerx)
    self.variacion_mouse_y = (self.mouse_coordenadas[1] - self.hitbox.centery)
    self.angulo_variacion = math.degrees(math.atan2(self.variacion_mouse_y, self.variacion_mouse_x))
    self.image = pygame.transform.rotate(self.base_jugador, -self.angulo_variacion)
    self.rect = self.image.get_rect(center = self.hitbox.center)
```

Acá se ve como en la primera línea obtenemos la posición x e y del mouse, luego como obtenemos el cateto opuesto y el cateto adyacente bajo los nombres de variables “variación mouse x” y “variación mouse y” para luego aplicar la función de tangente de la librería math. Por último, vemos de cómo se transforma a grados y se usa la función “rotate” para rotar la imagen.

### Bala (Bullet)

Ahora bien, recientemente acabamos de describir como generar la rotación del personaje sobre su propio eje. Si bien es muy importante esa implementación no ha de servir de mucho si no podemos lograr que los disparos que generan tomen el mismo comportamiento que la posición en el instante en que se realizan. Para ello he tenido que usar el ángulo tangente resultante usado para la rotación para aplicarlo a la “velocidad” de la bala, que no es otra cosa que un valor al cual en cada update() se le va sumando a la posición de la bala para moverlo en el espacio. Cuando un disparo es derecho y solo puede tener una dirección (como cuando se dispara de abajo hacia arriba) es sencillo trabajarlo ya que simplemente se deja la coordenada x fija y se le va sumando valor (velocidad) al eje y para generar el movimiento.

Lograr este efecto bajo la disponibilidad requerida en este video juego lo torna un poco más complejo ya que hay más variables a modificar. En primera instancia hay que variar los valores tanto de x como y. Además de eso también hay que lograr que cada disparo continúe la línea de dirección establecida por la posición de rotación al instante de ejecutarse.

Para lograr tal fin he usado el ángulo tangente que corresponde a la rotación obtenido de la instancia jugador según su posición de ubicación y la del mouse y la he usado para obtener tanto el seno para que baja dando la posición actualizada en cada paso de la coordenada Y, por otra parte, el coseno para que baja dando la posición actualizada de la coordenada X. A continuación, se puede ver la parte de código en la cual se implemente.

```
self.velocidad = VELOCIDAD_BULLET
self.velocidad_x = math.cos(self.angle * (2*math.pi/360)) * self.velocidad
self.velocidad_y = math.sin(self.angle * (2*math.pi/360)) * self.velocidad

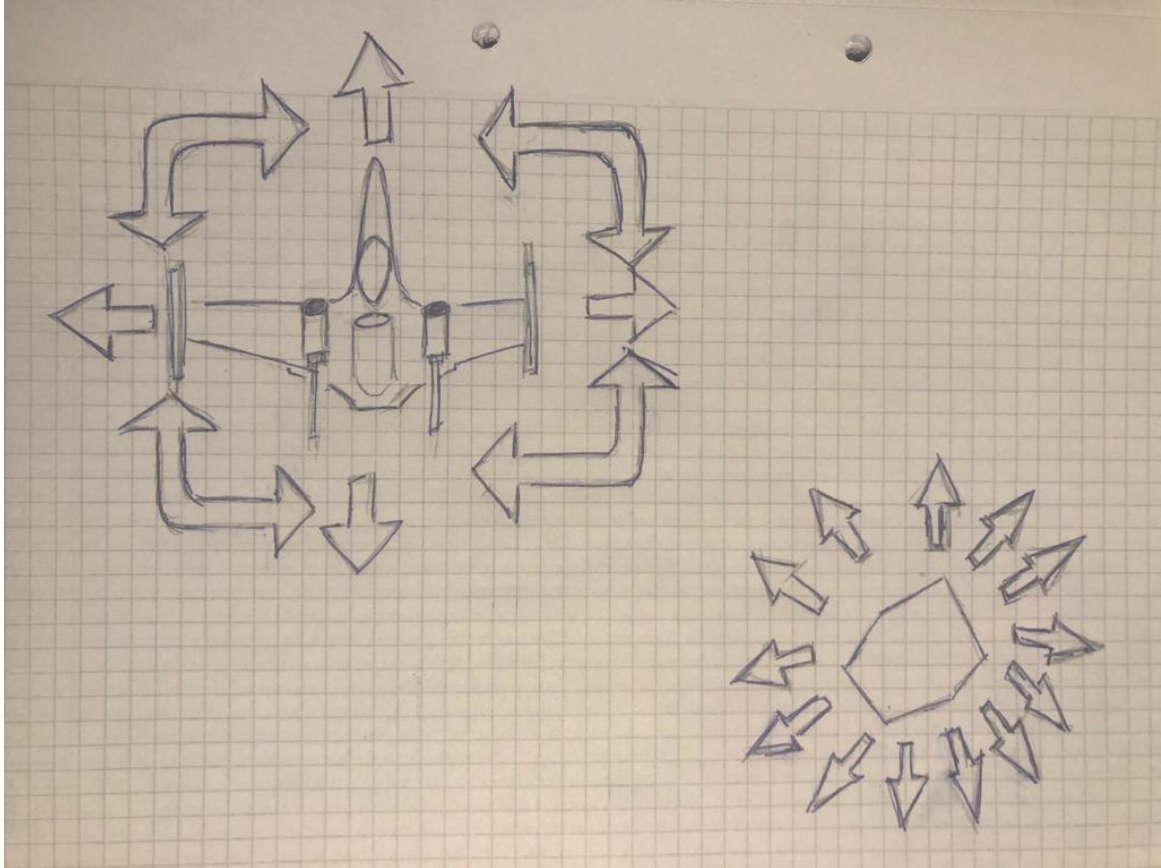
def movimiento(self):

    self.x += self.velocidad_x
    self.y += self.velocidad_y

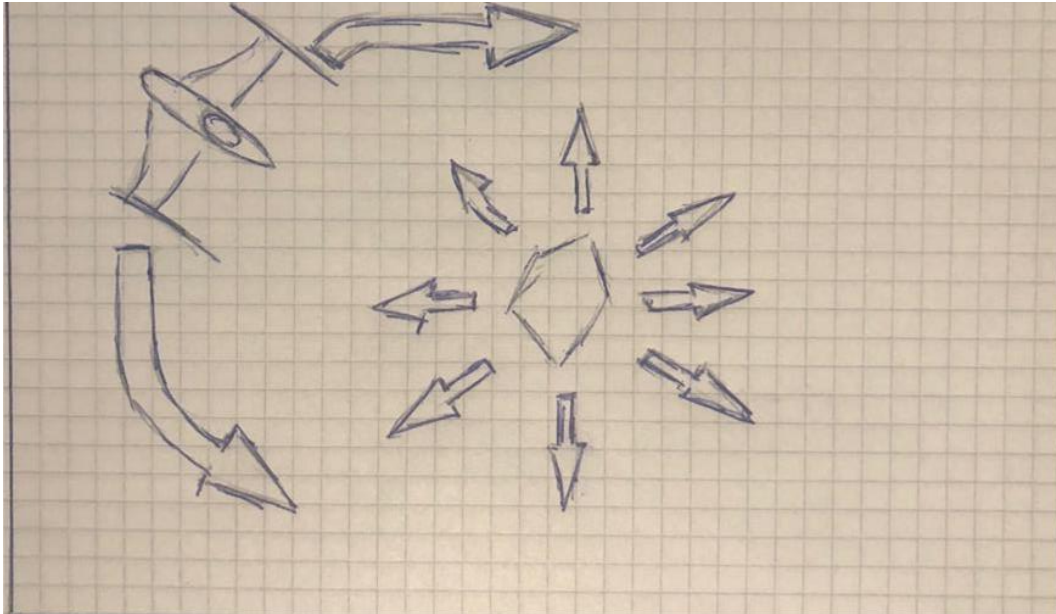
    self.rect.x = int(self.x)
    self.rect.y = int(self.y)
```

## Bocetos de disposición de la movilidad y distribución de los elementos en el espacio

Direccionalidad de los elementos:



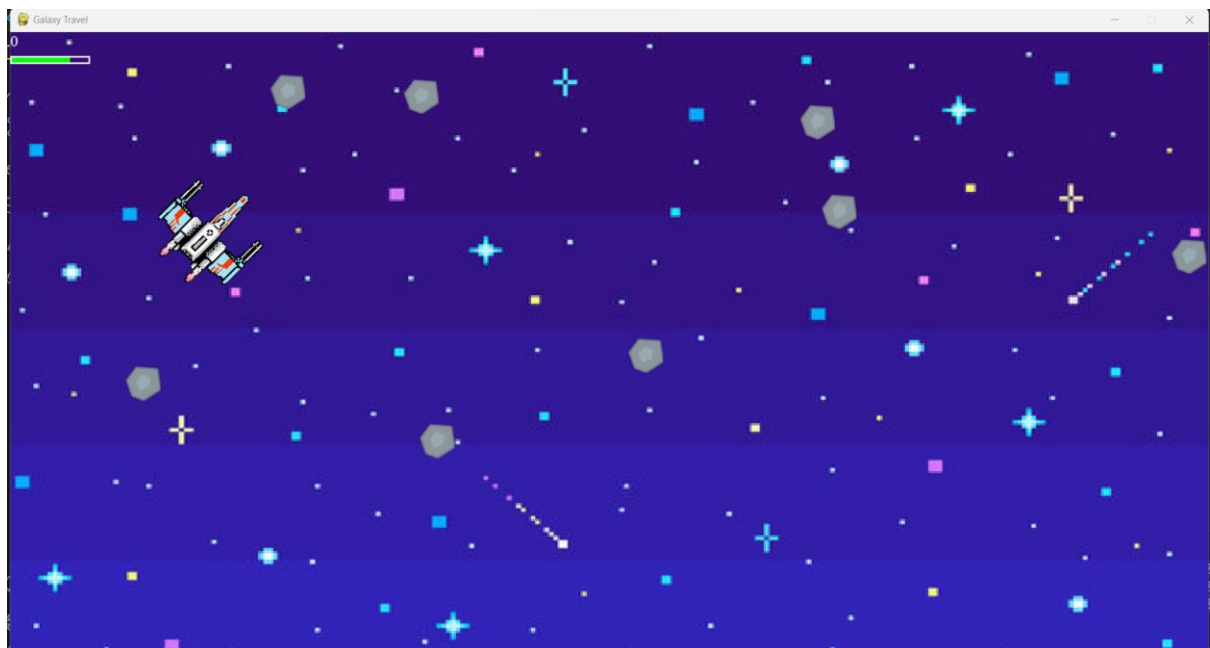
Disposición del Espacio:



## Estética

Por último, quiero hacer mención a la selección de la estética para el video juego.

La idea fue no generar una estética que demandará gran dificultad en su resolución pero que tenga una identidad y logre transmitir una idea clara al usuario. Por lo tanto, decidí basarme en sólo 4 elementos muy básicos y todos con la premisa de que sean estilo “pixel” para recrear una identidad retro.



# Reflexiones Finales

En principio puedo decir que estoy satisfecho con el juego realizado y su nivel. A continuación, hare un detalle de los puntos fuertes y débiles, como así también de las posibles mejoras que se podrían implementar a mi criterio en futuras versiones.

## Puntos Fuertes

Para mi uno de los puntos fuertes es el doble comando realizado con la nave, yo creo que lo hace bastante atractivo ya que demanda tanto la atención del usuario por su complejidad y a su vez también permite un amplio rango de movimientos y acciones.

Otro de los puntos fuertes me parece que es la aparición desde el centro de los asteroides con una direccionalidad random. Creo que el efecto está bien logrado.

## Puntos Débiles

Uno de los puntos que veo débiles en la jugabilidad del video juego es que no se termina recorriendo, en la mayoría de los casos, toda la pantalla. Por lo general se tiende a usar uno o como máximo dos de los cuadrantes de la pantalla.

Futuras mejoras que se pueden implementar

Me parece que agregar incentivos para que el jugador tenga que recorrer toda la pantalla puede ser un punto a mejorar en el nivel. Entiendo que eso se podría lograr colocando incentivos es rincones específicos distribuidos en el espacio y que estén en distintos cuadrantes de la pantalla a fin de obligar al jugador a recolectarlos para poder avanzar de nivel.